Music Perception
Spring 2002, Vol. 19, No. 3, 415–442
© 2002 BY THE REGENTS OF THE UNIVERSITY OF CALIFORNIA
ALL RIGHTS RESERVED.

How Jazz Musicians Improvise

P. N. JOHNSON-LAIRD

This article defends the view that theories of creativity should be computable and that only three sorts of algorithm can be creative. It proposes a central principle of *algorithmic demands* for jazz improvisation: a division of labor in terms of computational power occurs between the creation of chord sequences for improvisation and the creation of melodic improvisations in real time. An algorithm for producing chord sequences must be computationally powerful, that is, it calls for a working memory or a notation of *intermediate* results. Improvisation depends on the ability to extemporize new melodies that fit the chord sequence. The corresponding algorithm must operate rapidly in real time, and so it minimizes the computational load on working memory. The principle of algorithmic demands is supported by analysis and a computer model.

MUSIC is often improvised rather than composed. In the West, jazz improvisations are the most familiar example, but improvisation is common in many other cultures, and even in classical music of the European tradition. Listeners, for instance, were sometimes more impressed by Beethoven's extempore performances than by his compositions. For the psychologist, the topic of improvisation presents a unique challenge: an artist creates an original work in real time. Its creation depends on what is in the musician's head, although the process is often a collective effort by a group of musicians. The musicians' knowledge, mental processes, and musical skills somehow yield a complex piece of music that is admired (and paid for) by listeners.

Modern jazz derives from bebop, the style that Dizzy Gillespie, Charlie Parker, and others developed in the 1940s and that continues to be an influential style. A piece typically has the following form. It begins and ends with an ensemble statement of a composed melodic theme. Between these opening and closing statements, the musicians take turns to improvise melodic solos of several choruses. In early jazz, they merely embellished the melody of the theme (Schuller, 1968, pp. 73–74), but from Louis

Address correspondence to P. N. Johnson-Laird, Department of Psychology, Princeton University, Princeton, NJ 08544. (e-mail: phil@princeton.edu)
ISSN: 0730-7829. Send requests for permission to reprint to Rights and Permissions, University of California Press, 2000 Center St., Ste. 303, Berkeley, CA 94704-1223.

Armstrong and Sidney Bechet to Charlie Parker and his successors, jazz musicians have extemporized new melodies that fit the chord sequence of the theme. As a result of years of improvisatory practice, they can navigate their way through chorus after chorus of the chord sequence and create a seemingly endless series of melodies appropriate to its harmonic implications. They are able to produce such melodies, if need be, at the fastest rate that they can physically perform their instruments, namely, at a rate of 10 to 12 notes per second. Hence, improvisation does not seem to add to the cognitive load of playing the instrument. The chord sequences that modern jazz musicians prefer include the "12-bar" blues, original compositions of their own, and those that derive from popular songs, typically those of 32 measures in an AABA or ABAB form by such composers as George Gershwin, Jerome Kern, and Cole Porter. The rhythm section of piano, double bass, and drums provides the accompaniment for the horn players. The drums state the basic metrical pulse, usually a regular four beats to the measure, emphasizing the weak second and fourth beats, and playing rhythmic figures to stimulate the improvising soloist. The bass player improvises a bass line to the chord sequence and also helps to maintain the metrical pulse at a fixed tempo. The pianist improvises a version of the given chord sequence, varying the choice of chords and their voicings, and again providing rhythmic figures to help to create a feeling of "swing" (for complete transcriptions of extensive sections of such improvisations, see, e.g., Berliner, 1994).

Figure 1 shows the first eight measures of an improvisation by Bud Powell, one of the greatest pianists in modern jazz. It is a transcription that I made from a track on *Bud Powell, The Complete 1946–9 Roost/Blue Note/ Verve/ Swing Masters* (Definitive Records DRCD 11145). Powell was improvising to his version of the chord sequence of *Off Minor* composed by his mentor, Thelonious Monk. Powell, of course, played an accompaniment to

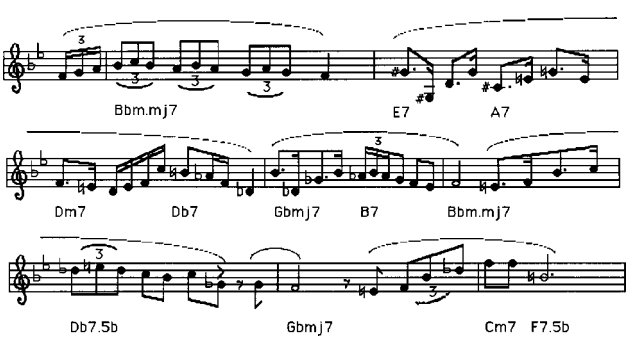


Fig. 1. Eight measures from the start of Bud Powell's improvisation on the chord sequence for Thelonius Monk's theme, *Off Minor*.

his own improvisation, but the figure merely presents the chord sequence in a conventional notation (see explanation later). The transcription does not make explicit the rhythmical quality of modern jazz. No good notation for it exists—musicians acquire the style by listening to virtuosos and seeking to emulate them, and they develop a discriminating ear for what "swings" and what does not. But, their skill is intuitive. They are not consciously aware of its underlying rhythmic principles.

The psychological question about modern jazz is: how do musicians improvise? That is, what are the mental processes that underlie their performance? They themselves can articulate only a limited answer, because the underlying mental processes are largely unconscious. If you are not an improvising musician, then the best analogy to improvisation is your spontaneous speech. If you ask yourself how you are able to speak a sequence of English sentences that make sense, then you will find that you are consciously aware of only the tip of the process. That is why the discipline of psycholinguistics exists: psychologists need to answer this question too.

In this article, I sketch a tentative solution to the riddle of improvisation. I presuppose that the process is computable. The point is not to replace musicians with computers, but rather to defend the view that a psychological theory of human creativity should be computable. That is to say, it should be possible to devise a computer program that models improvisation using the same sorts of principles that underlie human performance. There already exist programs that improvise jazz, but they tend to be exercises in artificial intelligence rather than cognitive modeling. Psychologists are a long way from implementing any computer program that can perform as skillfully as a human musician, and there is at least one major impediment to modeling improvisation computationally—an impediment to which we return later. So, the argument is one of principle rather than concrete proof. It is a development of an earlier attempt to cope with jazz improvisation (Johnson-Laird, 1991a).

The article begins with an analysis of creativity on the assumption that it can occur in lowly everyday thinking as well as in the highest domains of science and of arts such as jazz (Johnson-Laird, 1993). It follows from this analysis that only three sorts of *algorithm* could be creative, where an algorithm is a finite but effective set of instructions for carrying out a task. The paper then turns to modern jazz. Its main claim is that a principle of *algorithmic demands* governs improvisation: there is a judicious division of labor in terms of computational power between the creation of chord sequences for improvisation and the creation of melodic improvisations in real time. I argue that the creation of chord sequences depends on an algorithm that occurs over multiple stages of creation and critical evaluation, and that calls for a memory of intermediate results. In contrast, the impro-

visation of melodies depends on an algorithm that occurs in a single step and that does not call for any memory of intermediate results. The article illustrates how such an algorithm could work for the generation of rhythmic phrases and for the improvisation of pitches to convert them into melodic phrases. Finally, it describes a computer program illustrating these ideas.

A Computational Theory of Creativity

Many thinkers have supposed that creative processes are beyond the capacity of any machine. The great French mathematician Henri Poincaré (1929) wrote:

> Now we have seen that mathematical work is not simply mechanical, that it could not be done by a machine, however perfect. It is not merely a question of applying rules, of making the most combinations possible according to certain fixed laws. The combinations so obtained would be exceedingly numerous, useless, and cumbersome. The true work of the inventor consists in choosing among these combinations so as to eliminate the useless ones or rather to avoid the trouble of making them, and the rules which must guide this choice are extremely fine and delicate. It is almost impossible to state them precisely; they are felt rather than formulated. Under these conditions, how can we imagine a sieve capable of applying them mechanically?

Penrose (1989) has translated this view into modern terms: creativity depends on processes that are *not* computable. If his view is correct, then either certain mental processes cannot be emulated by any sort of computer or the theory of computability is wrong. It rests on the so-called Church-Turing thesis that any finite procedure that is effective for carrying out a task is computable (see, e.g., Boolos & Jeffrey, 1989). The only evidence for the Poincaré-Penrose hypothesis is its authors' intuitions. Others have different intuitions. Hence, whether or not creativity is computable is an open question. A sensible strategy is accordingly to assume that it is computable until one is forced to abandon this hypothesis. In contrast, a *theory* of creativity ought to be framed in a computable way. Early psychological theories of creativity are too vague to determine what sort of mental processes they postulate (e.g., Freud, 1908; Koestler, 1964; Mednick, 1962; Wallas, 1926). But, nowadays there is no good reason to propose theories that take so much for granted that they cannot be implemented in computer programs. If you proceed otherwise, you risk producing that most pernicious of intellectual artefacts, a theory that no one, not even its proponents, can properly understand.

What counts as a creative process? Science seldom advances by a priori definitions, but the following analysis at least provides a working definition that demarcates the field. It rests on five assumptions:

1. The result of a creative process is novel, at least for the individual doing the creating (Reber, 1985).
2. The result of a creative process may also be novel for society as a whole. But the mental processes underlying creativity are the same even if unbeknownst to the individual someone else has already had the same idea. Hence, novelty for society is optional rather than essential.
3. Imagination is more than calculation, imitation, or regurgitation. If you multiply two numbers together, the result may be a number that you have never thought of before. Yet, your response is not creative—even if your answer is *wrong*. Most people have the intuition that when they are creating something, such as improvising jazz or making an extempore speech, alternative possibilities occur at many points in the process. If they could relive the experience with no knowledge of their first effort, then they might take a different route the second time around. In computational theory, a machine with this property is known as *nondeterministic*. It can yield different outcomes when it is in the same internal state and has the same input, if any (see, e.g., Hopcroft & Ullman, 1979). No one knows whether human creativity is really nondeterministic, but it is convenient to adopt a *theory* of creativity that has this property. It allows for our ignorance, and, computationally speaking, it costs us nothing. In principle, anything that can be computed nondeterministically can also be computed deterministically.
4. Creativity satisfies preexisting constraints or criteria. Musicians create sonatas and symphonies, songs, and instrumental improvisations. They work within the constraints of a genre and of their own personal style. Their style may develop or change. And a revolutionary change may occur in a genre: in jazz, such a revolution occurred in the 1940s with the development of bebop. Even when a new genre is created, it too lies within the bounds of criteria, but it may face hostility and incomprehension from those who do not grasp its fundamentals. When I first heard bebop at the age of 12, I thought that the musicians were playing notes at random. I went to the piano and played at random; the result was not modern jazz. It has constraints. The society of musicians crystallized these constraints, which themselves were the consequences of previous creative processes. The individual

creator is not a closed system, but is influenced by mentors, collaborators, and leaders (see Simonton, 1984). The aesthetic values of a culture thereby exert an historical influence on the individual's creative processes, which, in turn, may contribute to the values that are passed on to the next generation.

5. Creations cannot be constructed out of nothing. There must be existing elements to provide the raw materials for even highly original works of art or science.

These five components make up the NONCE definition of creativity: creativity is *N*ovel for the individual, *O*ptionally novel for society, *N*ondeterministic, dependent on *C*riteria or constraints, and based on *E*xisting elements. In modern jazz, each improvisation is novel in the same way that most utterances in natural language are novel—the speaker has never uttered them before, and often neither has anyone else. As far as one can tell, improvisation is not deterministic, that is, it does not unwind like clockwork with only one choice for the musician at each point in the performance. There are often many choices. Yet, musicians do not have total freedom—at least in orthodox modern jazz. There are harmonic and rhythmic constraints on improvisations. Much of the hard work in learning to improvise consists in acquiring a tacit mastery of these constraints. The existing elements are the fundamental constituents of tonal music: pitches, timbres, chords, intensities, and durations.

Three Algorithms for Creativity

A consequence of the NONCE definition is that there are only three sorts of algorithm that could be creative. The first sort is neo-Darwinian, that is, analogous to the evolution of species according to the modern neo-Darwinian account. In this sort of algorithm, there are two stages in creativity: a generative one in which ideas are formed by an entirely arbitrary process working on existing elements, and then an evaluative stage that uses criteria to filter out just those results that are viable. Whatever survives, which may be little or nothing, can serve as the input to the generative stage again. The process can thus be repeated ad libitum with the creation from one iteration serving as the input for the next. Neo-Darwinist theories of creativity have often been proposed by psychologists (e.g., Bateson, 1979; Campbell, 1960; Skinner, 1953). An individual produces a variety of responses, and the contingencies of reinforcement, or some other criteria, select those that are viable and extinguish the remainder. It is crucial to distinguish between a single operation of a neo-Darwinian procedure and its iteration, which is much more powerful (Dawkins, 1976). Evo-

lution is thus an archetypal *recursive* process: it applies to its own successful results. It is mimicked by the "genetic algorithms" developed by Holland and his colleagues for finding optimal solutions to problems (e.g., Holland, Holyoak, Nisbett, & Thagard, 1986).

A neo-Darwinian algorithm is grossly inefficient, but it is the only feasible one if the generative process cannot itself be guided by criteria—an assumption underlying the evolutionary synthesis of genetics and natural selection (see Mayr, 1982, p. 537). Yet, if criteria are used in the evaluation of possibilities, why not use them instead to constrain the generative stage? Why not, indeed, for unlike species, ideas could evolve using this process. The second sort of algorithm is *neo-Lamarckian* in just this way. All the criteria acquired from experience govern the generative stage—by analogy with Lamarck's theory of evolution (see Mayr, 1982, p. 354). If an individual has mastered a set of criteria that suffice to guarantee the viability of the results, then the generative stage will yield a small number of possibilities, all of which meet the constraints of the genre. The algorithm will be efficient, and it will never produce hopeless results. But, if all the individual's criteria are used to generate a result, by definition nothing is left for its evaluation. Granted that creation is not deterministic, there will be certain points where the criteria allow more than one possibility, and so the only way to choose among them will be nondeterministic, for example, by making an arbitrary decision. The algorithm has just two stages with no need for recursion: (1) the generation of possibilities according to criteria, and (2) an arbitrary selection, where necessary, from amongst them.

The third sort of algorithm is a compromise. The generative stage uses some criteria like a neo-Lamarckian algorithm, and the evaluative stage uses some criteria like a neo-Darwinian algorithm. In other words, the initial generation of possibilities under the guidance of some criteria leaves something to be desired, and so the individual applies further criteria to evaluate the results. They may need further work, and so the process recurses through multiple iterations. Many creative individuals do indeed work extensively over the results of their earlier efforts, revising and revising and revising. They may try out novel ideas in an experimental way. They are using a multistage procedure. But, since they are applying criteria at each stage, why don't they apply all of these criteria straightaway in the very first generative stage? Why the need for a time-consuming division of labor over several stages? From a computational standpoint, it would be more efficient to apply all the criteria in the generative stage in a neo-Lamarckian way. It is paradoxical for individuals to waste time making an inadequate attempt if they have the ability to perceive its inadequacy and to set matters right.

The resolution of the paradox may lie in the way the mind works. Knowledge for generating ideas is unconscious and embodied in procedures. It is

knowledge of *how* to do things. But, knowledge for evaluating ideas can be conscious and embodied in beliefs. It is knowledge *that* something is the case. This dissociation also resolves another puzzle, one that Perkins (1981, p. 128) refers to as the fundamental paradox of creativity: people are better critics than creators. Criticism can be based on conscious knowledge that is acquired easily; whereas the generation of ideas is based on unconscious knowledge acquired only by laborious practice in creating. Hence there are two stages in many sorts of creation: a generative stage and an evaluative stage. And hence the greater ease of criticism over imagination. In creation, no substitute exists for a period of apprenticeship. You learn by imitating successful creators and by trying to create for yourself in a particular domain. Only in this way can you acquire the tacit criteria of a genre or paradigm. It follows that there is no general recipe for enhancing your creativity across all domains. Creation is specific to a particular domain of expertise.

Algorithms for Jazz

The cognitive problem for jazz musicians is to create a novel melody that fits the harmonic sequence and the metrical and rhythmic structure of the theme. The musicians must therefore be highly familiar with the chord sequence, use their working memory to keep track of where they are in that sequence and to register what other musicians are playing, and generate and execute sequences of notes in pleasing musical phrases. The improviser of modern jazz is therefore akin to what is known as a "transducer" in automata theory—a transducer that takes as input step by step a chord sequence from long-term memory and a perception of the accompaniment, and that produces an output consisting of an improvised melody. Virtuosos seldom, if ever, play wrong notes: each phrase dovetails with the harmonic implications of the chord sequence. Evidently, jazz musicians have an internal representation of the musical criteria governing the generation of melodies.

The bottleneck in improvisation, as in cognition in general, is the limited processing capacity of working memory. Computational power, that is, what a system is able in principle to compute, depends on a working memory for the intermediate results of computations. Working memory holds these intermediate results until they are needed again, and so in computational theory there is a hierarchy of automata that are able to compute ever more powerful results because of increases in the effective capacity of their working memory. Likewise, there is a matching hierarchy of grammars—the so-called Chomsky hierarchy—that can characterize the performance of these automata (see Hopcroft & Ullman, 1979). The simplest device capable of

generating infinitely many different expressions is a so-called *finite-state* automaton, which makes no use of working memory over and above the finite number of internal states into which it can enter. A simple example is the automaton required for the addition of two binary numbers. It needs no working memory: it is either in a state in which there is a carry, or not. But, the automaton required for binary multiplication needs a working memory to store the results of intermediate products. Multiplication calls for more computational power than addition.

The NONCE definition of the previous section implies that the three sorts of algorithm are exhaustive (Johnson-Laird, 1993). Two of the three sorts are used to solve the problems of improvisation and to divide its cognitive demands between working memory and long-term memory. This division favors the generation of musically interesting ideas at a rapid rate in real time. The central thesis of this article is accordingly that improvisation depends on a principle of *algorithmic demands*:

> The composition of tonal chord sequences depends on a multistage algorithm that requires a working memory (or, equivalently, a notation) for intermediate results, whereas the tacit procedures for the improvisation of melodies depend on a neo-Lamarckian algorithm that requires no working memory for *intermediate* results.

It is easy to misunderstand the nature of this claim. A reviewer wrote that it negates all the analyses by Pressing (1987), Schuller (1989), Berliner (1994), and others, showing "the long range motivic and developmental structural order in jazz solos." In fact, the claim has no such implication. Jazz musicians can generate long-term relations in their improvisation without using working memory. They can make repeated use of a motif or a phrase throughout an improvisation because it is in long-term memory. It may be an habitual phrase. But, when musicians extemporize a striking phrase, they are likely to store it in long-term memory and perhaps to improvise variations on it. The phrase accordingly becomes part of the input to the transducer from long-term memory. There is no need for a working memory of intermediate computational results.

What would call for such a working memory? On the one hand, it would be needed for a highly constrained musical language that, for instance, demanded that all improvised melodies took the form of exact palindromes such as B♭ACE♭DDE♭CAB♭, for which there was no bound in length (see, e.g., Hopcroft & Ullman, 1979). On the other hand, working memory would be needed if dependencies in melodies, or the mental representations required to generate them, were as constrained as those in natural language. A simple finite-state device suffices, in principle, for the agreement between the number of the subject noun-phrase and the number of its corresponding verb, namely, whether they are both singular or both plural.

But, as Chomsky (1957) pointed out, an indefinite amount of material can intervene between subject and verb, and this material may include relative clauses that themselves call for number agreement, for example, *The woman the pupils admire likes running*. The utterance of embedded clauses, such as *the pupils admire [the woman]*, calls for a working memory to keep track of unfinished business, such as the number of the noun phrase *the woman*. This load on working memory has a detectable effect on other skills, such as tracking a visual target (Power, 1986). When violations of these constraints occur in spontaneous speech (e.g., a subject and verb fail to agree in number), linguistic judgment tells us that an error has been made. The sentence is not grammatical. To make a comparable case for structure in improvised melodies, it would be necessary to establish that there are long-term musical dependencies, that these dependencies can occur at the very least embedded one with another to an arbitrary degree, and that violations of the dependencies, if they were to occur, would count as musical errors. No such case has ever been made: structural relations embedded within one another to an arbitrary degree do not occur in improvised melodies in an inviolable way. The long-range dependencies that jazz theorists have observed can be generated using a finite-state transducer without a working memory for intermediate results. The principle of algorithmic demands is therefore consistent with current musical analyses.

The motivation for the principle of algorithmic demands is simple: if musicians had to calculate intermediate results—like the partial products of long multiplication—the process of improvisation would be slowed down. It would take too long to figure out what to play. In fact, jazz musicians can improvise at the fastest speeds that it is physically possible to play a musical instrument. This phenomenon provides prima facie support for the principle. In what follows, I seek further corroboration for the principle. I examine the composition of tonal chord sequences suitable for jazz improvisations, the improvisation of rhythmic phrases within a metrical framework, and the creation of melodies by fitting pitches to these phrases.

Tonal Chord Sequences: A Multistage Algorithm

Tonal music is in a key, that is, a subset of notes (a scale) in which one note (the key note or "tonic") serves a central function, and other notes are related to it in a space of three principal axes: octaves, major thirds, and fifths (see, e.g., Krumhansl, 1990; Longuet-Higgins, 1987, Sec. II). A chord sequence depends on two components: the sequence of the roots of the chords, and the particular intervals that occur in each chord. Although much has been written about tonal chord sequences, many theories have fallen into the same traps as early psychological theories of creativity. On

the one hand, they are vague—too vague, at least, to be modeled in computer programs; on the other hand, they lack sufficient computational power to do the job. Thus, from Rameau (1722) to Forte (1979), theorists have described chord sequences in informal ways equivalent to the generation of sequences without the use of working memory. But, as we will see, the most interesting chord sequences cannot be composed in this way.

Most jazz musicians have a conscious knowledge of the main sorts of chords and of the sequences of chords of the themes in their repertoire. Any chord, such as F dominant seventh can be realized in many different ways, although most of them will include at least one occurrence of the notes F, A, C, and E♭. In modern jazz, the chord may have additional notes, such as sixths, ninths, elevenths, and a minor third to give it a "blues" sound, and it may occur in an inversion in which, say, E♭ is the root and F does not occur. Granted such variations, there are six principal sorts of chord in modern jazz, which may occur on any root. Musicians use a variety of names and notations for these chords (see Witmer, 1988). They are stated here with symbols commonly used in jazz:

Chord notation	Notes in the chord with root C
Cmj7	C, E, G, B
C7	C, E, G, B♭
Cm7	C, E♭, G, B♭
Cm7.5b	C, E♭, G♭, B♭
Cm.mj7	C, E♭, G, B
Cdm7	C, E♭, G♭, A

Of course, other sorts of chord occur, but these six and variants of them appear to be the most frequent in modern jazz.

An illustration of a modern jazz chord sequence is Thelonius Monk's aptly named *Off Minor*, which hovers around a minor key. It has an AABA 32-measure structure, and its A section chords—in Bud Powell's variant—are shown here with Roman numerals designating the roots of the chord, where I is the keynote, V its dominant, and so on:

Im.mj7	bV7 VII7	IIIm7 bIII7	bVImj7 bII7
Im.mj7	bIII7.5b	bVImj7	IIm7 V7.5b

where "5b" indicates that the dominant seventh chord is played with a flattened fifth. The actual choice of voicings for the chord is extemporized by the instrumentalists providing the accompaniment, and they may depart from the sequence in various ways, for example, they may interpolate passing chords.

Rules that correspond to a finite-state device (with no working memory) can capture only a simple binary structure. Such rules, though often implicit in theories of harmony, assign the sort of structure shown in Figure 2.

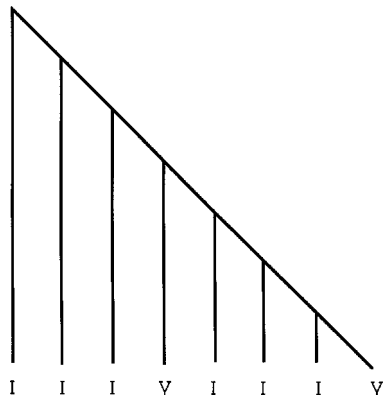


Fig. 2. A binary structure assigned to a chord sequence created without the use of working memory.

This simple sequence of bifurcations is at odds with the real structure of the sequence: it does not capture the salient division into two sets of four measures each. Such a division calls for more powerful rules, which correspond to the use of a working memory for intermediate results. These rules can specify the cadences that are basic to tonal chord sequences, such as:

1. Opening cadences from the tonic to a new root.
2. Closing cadences back to the tonic.

The new root in an opening cadence is often the dominant, but other roots are possible. These cadences underlie the set of rules presented in Table 1. They generate the *underlying* tonal chord sequences of the sort that occur in the A section of an AABA theme. Table 2 presents some examples of their output, and Figure 3 shows the underlying chords that they assign to *Off Minor*. How does such a simple underlying structure get transformed into the actual sequence of chords in the theme?

A common phenomenon in jazz is that the same underlying chord sequence can occur in different variations. To capture these variations, Steedman (1982) proposed a set of rules rooted in Longuet-Higgins's (1962) theory of tonality. Steedman's rules allow various substitutions and interpolations well-known to jazz musicians. They include, for example, a rule that allows a chord, X, which precedes a seventh, for example:

X F7

to be replaced by a minor seventh chord on the dominant of the chord it precedes:

Cm7 F7

More recently, Steedman (1994) has shown how rules for substitution and interpolation can be replaced by the use of a single so-called "categorial"

TABLE 1

A *Context-Free* Grammar for Generating 8 Measure Tonal Chord Sequences

Eight measures	→	First four	Second four
First four	→	Opening cadence	Opening cadence
	→	Opening cadence'	Opening cadence
Second four	→	Opening cadence	Middle cadence
		Opening cadence'	Closing cadence
		Middle cadence	Middle cadence
		Middle cadence	Closing cadence
Opening cadence	→	| I	| I |
	→	| I	| V |
Opening cadence'	→	| I	| III |
	→	| I	| IV |
Middle cadence	→	| I	| V |
	→	| V	| V |
Closing cadence	→	| I	| I |
		| IV	| I |
		| V	| I |

The rules all concern variants on opening cadences (from tonic to dominant). The arrows signify that the symbols on the left can be realized by the symbols on the right.

TABLE 2

Examples of the Chord Sequences Generated By a Program Using the Grammar in Table 1

| I	| V	| I	| V	| I	| IV	| I	| I	|
| I	| IV	| I	| V	| I	| III	| V	| I	|
| I	| III	| I	| V	| I	| IV	| I	| I	|
| I	| V	| I	| V	| I	| V	| V	| I	|
| I	| I	| I	| I	| I	| V	| IV	| I	|

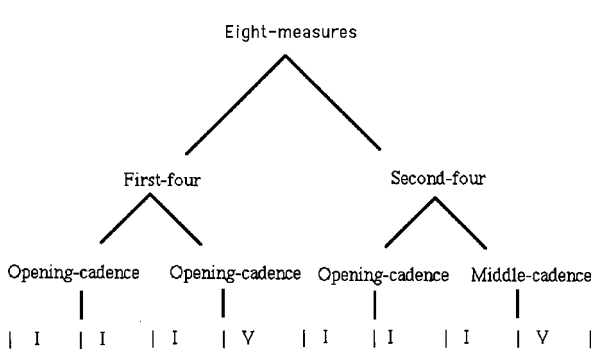


Fig. 3. The structure of the *underlying* tonal chord sequence for *Off Minor*.

grammar, which contains a lexicon defining the syntactic properties of different chords, e.g., a minor seventh chord is a function that returns a minor seventh given that it is followed by seventh chord a fourth up. The crux for the present article, however, is that the creation of chord sequences demands a working memory for intermediate results.

Consider how a musician might substitute and interpolate chords in order to derive the actual chords of Monk's *Off Minor* from its underlying tonal cadences (produced by the grammar in Table 1). The first four measures in the underlying sequence in Figure 3 are:

| I | I | I | V |

A sequence of substitutions, especially of chords a flattened fifth away from the originals, and interpolations working backwards according to the "cycle of fifths," yield a most unusual sequence of chords, characteristic of Thelonius Monk:

A substitution:				| bII7 |
An interpolation:				| bVImj7 bII7 |
An interpolation:			| bIII7	| bVImj7 bII7 |
An interpolation:			|bVIIm7 bIII7	| bVImj7 bII7 |
A substitution:			| IIIm7 bIII7	| bVImj7 bII7 |
An interpolation:		| VII7	| IIIm7 bIII7	| bVImj7 bII7 |
An interpolation:		| bV7 VII7	| IIIm7 bIII7	| bVImj7 bII7 |
A substitution:	| Im.mj7	| bV7 VII7	| IIIm7 bIII7	| bVImj7 bII7 |

The second four measures in the underlying sequence are the same as the first four (see Figure 3). The following manipulations occur:

A substitution:				| V7.5b |
An interpolation:			| II7	| V7.5b |
A substitution:			| bVImj7	| V7.5b |
An interpolation:		| bIII7.5b	| bVImj7	| V7.5b |
A substitution:	| Im.mj7	| bIII7.5b	| bVImj7	| V7.5b |
An interpolation:	| Im.mj7	| bIII7.5b	| bVImj7	| IIm7 V7.5b |

There are constraints on the acceptable interpolations and substitutions. The substitution of one seventh by another a flattened fifth away, for instance, is permissible if the next a chord is a fifth away (or a substitution for such a chord, as in the second substitution in the preceding example). Likewise, the interpolation of chords according to the cycle of fifths should not continue back to the point where an opening tonic in a cadence is eliminated.

In modern jazz, modulations from one key to another occur either from one major section of a chord sequence to another or else within such sections. A typical example of modulation between sections occurs in Clifford Brown's theme *Joyspring*, which repeats the first eight measures as the second eight measures, but modulated up by a semitone. Modulation between

sections often occurs from the main eight measures to the so-called "bridge" section in the AABA form, that is, the first eight measures (A) are repeated, and then lead to a bridge (B) in a new key, prior to the final reprise of A in its original key. Modulation is also frequent within sections. Indeed, the bridge may be in no fixed key at all, but simply modulate through the cycle of fifths. For example, a bridge based on Gershwin's *I Got Rhythm*, where the key of the main eight measures is Bb major, proceeds through the cycle of fifths:

| D7	| D7	| G7	| G7	|
| C7	| C7	| F7	| F7	|

There seem to be no constraints on modulation: a theme can modulate to any new key (see also Cork, 1988). Modulations are carried out either by a direct change to a new tonic or else by interpolating chords backwards from the new tonic according to the cycle of fifths. These interpolations can be handled by the sorts of rules invoked by Steedman (see preceding), and so the only rule for modulation is one that signifies a direct change in the tonic.

The computational rules for creating a tonal chord sequence call for the use of working memory. Pianists can, of course, improvise chord sequences. But, one corollary of the principle of algorithmic demands is that if they were to improvise them at the fastest speed with which they can play the piano, then they are unlikely to produce satisfactory chord sequences. Indeed, if the computational demands of generating interesting chord sequences were no greater than those of generating interesting melodies, then we cannot explain why most modern jazz improvisations are based on composed chord sequences, or why solo musicians improvise melodies to composed chord sequences rather than improvise chord sequences to composed melodies.

Composers do not need to rely on their working memories in order to create chord sequences. They can use musical notation instead. Likewise, in other arts, an unfinished picture, text, or poem, is itself a "memory" of an intermediate result. In practice, the composition of chord sequences for jazz depends on a multistage procedure, that is, sequences are rarely improvised in public performance or composed in one step in their final form. A sequence is normally refined over several stages of work. The composer tries one idea and then revises it; and a chord sequence in jazz, such as the 12-bar blues, may even evolve over several generations of musicians. Indeed, whenever a creative process calls for working memory, it is likely to depend on a multistage procedure. The greater degree of computational power has an aesthetic advantage. The resulting chord sequences can have a more interesting structure than would otherwise be possible. Because a jazz musician commits these chord sequences to long-term memory, they can enhance the quality of an improvisation without complicating the pro-

cess of improvisation itself. That, in essence, is the principle of algorithmic demands.

The Improvisation of Melodies: A Neo-Lamarckian Algorithm

The improvisation of a melody in jazz, like other extempore performances, occurs rapidly and with no opportunity to try out different possibilities and to choose only the best of them. From a computational standpoint, it depends on a neo-Lamarckian algorithm in which all the criteria for the music are used in generating a melody. If the criteria allow more than one possibility, then an arbitrary choice is rapidly made from among them. The principle of algorithmic demands implies that the algorithm should not rely on working memory. That is, it should not call for the representation of intermediate results, so that the musician can create music as rapidly as possible.

One view of jazz improvisation is that musicians string together a sequence of motifs—"licks" as they used to be called—modified to meet the constraints of the chord sequence. As Ulrich (1977) wrote: "Sequences of motifs are woven together to form a melody. Rather than constantly inventing new motifs, the musician modifies old ones to fit new harmonic situations." A similar idea underlies Levitt's (1981) program for improvising jazz melodies. And there are books containing sets of licks for neophytes to commit to memory to help them to learn to improvise. Yet, the motif theory cannot be the whole story. Someone had to invent the motifs in the first place, and in so doing they could not have been regurgitating them from memory. Musicians often reuse certain phrases, rhythmic patterns, and melodic contours, but they also play many phrases that are novel. Indeed, the task of committing to memory a large number of motifs and constructing complete solos out of them would be altogether impracticable. For experienced musicians, it is much easier to make up new melodies.

An improvised jazz solo is made up from phrases, and a note in a phrase has five main components: a pitch, which in jazz may be bent; an onset time with respect to the metrical structure of the measure; a duration; an intensity, which again may change during performance; and a manner of articulation, such as staccato, legato, slurred, or ghosted. A phrase may also contain rests, that is, silences that play a particular role in its musical shape. The specification of a phrase is complete when every note and rest in the phrase has been defined for all of these components. Undoubtedly, however, a musician is mainly concerned with two tasks: the generation of a rhythmic pattern, that is, a sequence of onsets of notes and rests, and the generation of a correlated sequence of pitches for the notes in the phrase. These two tasks are not independent of one another, but the following

sections consider them separately in order to examine the principle of algorithmic demands.

Rhythm and Meter

Jazz is a metrical music. The function of meter is organizational: it provides the framework for rhythm (see, e.g., Johnson-Laird, 1991b; Lerdahl & Jackendoff, 1983). Hence, meter is more than the number of beats in the measure: each beat can also be subdivided in a regular way. Longuet-Higgins and Lee (1984) have shown that such groupings can be captured by rules that make explicit the structure of measures. A good example is the contrast between two beats to the measure that are each subdivided into three, and waltz time with three beats to the measure each subdivided into two. One measure in these two meters contains the same number of units, but their structures differ. A measure of the first sort has the following structure:

(((♪) (♪) (♪)) ((♪) (♪) (♪)))

A measure of waltz time, however, has a different grouping of the same six units:

(((♪) (♪)) ((♪) (♪)) ((♪) (♪)))

The proportion of the measure within a matching pair of parentheses indicates its metrical importance, for example, the first beat in the measure is the most important unit, because its left-most parenthesis is matched by one at the end of the measure.

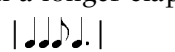
How much working memory, if any, is required for musicians to improvise in a metrical framework? The maximum number of beats that can be apprehended as making up an undivided measure is probably 7 ± 2 (cf. Miller, 1956), and beyond that the beats are divided into groups. In realizing the first beat of the measure, musicians have to retain only a representation of the remaining beats in the measure. It follows that a memory buffer of a fixed size will suffice to generate, or to perceive, a metrical structure.

As many theorists have pointed out (e.g., Povel, 1984), the critical feature of a rhythm is the sequence of onsets of its notes. Hence, if you clap the rhythm of a familiar piece, then listeners will be able to identify it. Clapping, of course, provides information only about onset times. The role of meter in the perception of music can be shown in a simple experiment (carried out by my former colleagues Jung-Min Lee and Malcolm Bauer). The experimenter counts, "1 2 3 4," in a regular way to establish a meter, and then claps the following rhythm in the same tempo:

| ♩♩♩♩ |

Listeners judge all four notes to be of the same duration. If, instead, the experimenter claps:

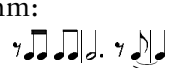
then the listeners judge the last note to be shorter than the others. These judgments are striking because the claps are all of the same brief duration. So, why is the last note in the first case judged to be longer than the last note in the second case? The answer must be that listeners perceive both rhythms as having a meter of four beats to the measure, and that they tacitly assume that there will be a note on the first beat of the next measure. The interval from the onset of the last clap to the onset of this imagined clap is indeed longer in the first case than in the second. When musicians served as participants in the experiment and were told not to try to notate the sequences in their mind's eye, they tended to judge that the last note in the following sequence ends with a longer clap:



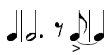
In short, listeners assume unconsciously that the onset of the next clap, even though it is purely imaginary, will occur on the first beat of the next measure. These judgments demonstrate the cognitive reality of metrical structure.

At its highest level of organization, a jazz improvisation is made up from phrases, which can vary in length from brief interpolations (see the phrase ending in the penultimate measure of Figure 1) to phrases that spread over several measures. Musicians aim for variety, but no long-range musical rules appear to govern the structure of an improvisation above the level of individual phrases. For the reasons described earlier, a demonstration of their existence would have to show that violations of long-range structural dependencies count as musical errors akin to grammatical errors. The things that hold a solo together are instead the use of simple motifs and the repeating harmonic sequence of the theme.

What counts as a phrase? For the player of a wind instrument, a phrase is typically played in a single breath. Cognitively speaking, listeners need to get their breath back too. Hence, a phrase is analogous to a sentence. It allows time for the listener to make musical sense of the events that precede its ending. If you listen to music in a familiar genre, you will usually have a clear intuition about where one phrase ends and another starts. The boundary between them is demarcated by several cues. The main cue is rhythm. There is a longer than average interval from the onset of the last note in a phrase to the onset of the first note in the next phrase. That is, the interval between phrases is longer than the intervals within phrases. But other cues can matter too. The opening phrase of *Walkin'*, a composition by Miles Davis, has the following rhythm:



The phrase ends, not with the first note in the second measure, but with the accented syncopation at the end of that measure, which is tied through to the next measure. Onset is a minimal cue here, but the tonality of the piece is decisive: the first note in the second measure is on the augmented fourth of the key—an unlikely note with which to end a phrase—and it leads to the fifth, which is the syncopated note that ends the phrase. An alternative version of the theme is played as:

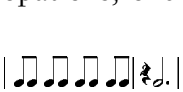


This version bears out the hypothesis to which we now turn: rhythmic phrases come in families that derive from underlying prototypes.

According to a theory of rhythms (Johnson-Laird, 1991b), the prototype of a rhythm depends on three distinct categories of musical event. First, and most important, is a syncopated note, which has an onset in an unexpected place—as though it has been displaced and starts in anticipation of its proper place in metrical structure. More precisely, following Longuet-Higgins and Lee (1984), a syncopated note has an onset on a metrical unit of lesser importance than one that intervenes before the onset of the next note. No meter, no syncopation. And, a syncopated note can be perceived as such only by the absence of a note on the next metrical unit of greater importance. The importance of a metrical unit, as we saw earlier, depends on the proportion of the measure within its parentheses. A measure in common time has the following structure at the level of eighth notes:

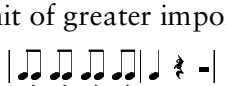
((((♪) (♪)) ((♪) (♪))) (((♪) (♪)) ((♪) (♪))))

Hence, the first of these units is of the greatest importance, because its leftmost parenthesis includes the whole measure. The subtlety of syncopation can best be grasped from a pair of contrasting examples. The following phrase contains two syncopations, one on the penultimate note and one on the final note:



but the next phrase has no syncopations:

In jazz, there is another sort of syncopation outside the preceding definition. A note can receive an emphasis—an accent—more appropriate to a note on the next metrical unit of greater importance, for example:

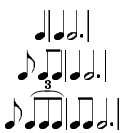


In reviewing this article, Jeff Pressing argued to the contrary that meter is not necessary for syncopation. He wrote: "Pulse alone (which is not meter) is sufficient to allow syncopation effects, as shown in work by Ornette Coleman, the Miles Davis free rock period, and others." But, as Kernfeld (1988, p. 87) has argued: "Syncopation depends for its effect on a persist-

ing sensation of the beat …; unless the beat is preserved in another voice in the ensemble or is swiftly reasserted, the listener loses his consciousness of the metrical framework, or even of the beat itself, and the syncopated pattern ceases to be perceived as such."

The second most important event in defining a prototype is a note with an onset on the beat, or at slow tempi on a major subdivision of the beat.

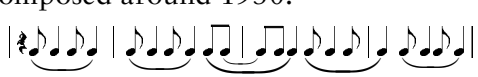
The third and least important event includes all other possibilities. The category contains rests, notes that last through the beat, and notes with onsets that do not occur on a beat. Phrases can be classified into separate families that each have the same underlying prototype in terms of the three sorts of event: *syncopations*, *notes* commencing on the beat, and the miscellaneous events of the *other* category. Each of the following rhythms is a member of the same family of the sequence, *other note note syncopation*, where the initial *other* event is optional:



As these examples show, the occurrence of *other* events after a *note* on the beat does not affect the prototype. The last of these examples is the final phrase in the fragment from Bud Powell's solo (see Figure 1).

Although families of rhythms can be formally derived from underlying prototypes, jazz musicians themselves are likely to have access to a single set of principles for creating phrases. Phrases have no structural dependencies over unlimited numbers of intervening notes. They can therefore be created without the use of a working memory for intermediate results. A minimal memory buffer suffices both to generate them and to place them in relation to metrical structure. One other principle simplifies the creation of rhythms. In jazz, an acceptable rhythmic pattern can start anywhere within the metrical structure of a measure provided that *syncopations* and *notes* do not change their status. As an example, consider the following two phrases:

Their rhythms are identical, but the second phrase is displaced one beat forward in the measure. The use of the same rhythm in different parts of the measure is a typical element of jazz. Indeed, the entire eight measures of the main A section of Monk's composition *Epistrophy* consists of the repetition of the two phrases just shown. The same device occurs in the following four measures leading into the final statement of Duke Ellington's *Old Man Blues*, composed around 1930:

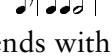


The same syncopated rhythm is repeated five times, starting first on the second beat of a measure, and cycling through the four beats of measures, until it starts again on the second beat of a measure.

Although rhythmic phrases can be generated without any major demands on working memory, perhaps the simplest way to describe the underlying principles is in terms of how an idealized musician might *learn* to improvise rhythms. The musician's first task it to learn to create rhythms according to simple prototypes. The following schema starts with an optional event in the *other* category, such as a pick-up note, and continues with a finite number of notes starting on the beat (the asterisk signifies any number of notes):

1. (Other) Note*

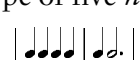
There is an additional convention that the last note in a phrase is followed by an interval that is longer than the onset intervals within the phrase. Schema 1 allows, for example, the prototype: *other note note note*:



A contrasting set of prototypes ends with a syncopation:

2. (Other) Note* Syncopation

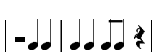
The schema generates a prototype of five *notes* followed by a syncopation:



Another sort of syncopation occurs at one level down in the size of the metrical unit:

3. (Other) Note* Syncopation: 1/8

The prototype with five *notes* before the *syncopation* yields the following case:

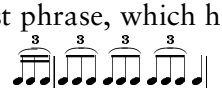


A syncopation can occur early in a phrase too:

4. (Other) Note Syncopation: 1/8 Note*

The prototype with five *notes* after *other* and *syncopation* yields the phrase:

When a musician has mastered a set of prototypes, the next task is to acquire the ability to realize each prototype in a large variety of different ways. These ways are illustrated by the six phrases in Bud Powell's improvisation in Figure 1. The first phrase, which has the rhythm:



is a realization of Schema 1 with four notes. Likewise, the second, third, and fifth phrases are instances of the same schema with eight, five, and one *note* respectively; the sixth phrase is an instance of schema 2, and the fourth phrase is an instance of schema 3.

In sum, musicians may learn the principles for improvising rhythmic phrases in three distinct systems: a set of prototypes, a set of principles for realizing them in various ways, and a system for timing the notes in a way that swings. Yet, the end result is probably a single set of unconscious procedures in a neo-Lamarckian algorithm that requires no working memory for intermediate results.

Melody and Pitch

The preceding section explained how musicians could create rhythms, but how do they add pitches to them to convert them into improvised melodies? One might suppose that the musician merely chooses any note from among those in the current chord in the harmonic sequence. This procedure is hopeless. The resulting melody—as a computer program showed (Johnson-Laird, 1991a)—leaps wildly around from a low note to a high note in a most unmelodic way. Moreover, musicians generate melodies that include so-called passing notes, which are not in the current chord. In fact, the possible notes in an improvisation are governed by two main constraints.

The first constraint is that the current chord in the harmonic sequence suggests a particular scale from which the notes to be improvised should be drawn. This idea is well-known to practitioners (see, e.g., Cork, 1988). The scale depends not only on the nature of the chord, but also on its context. A good example occurs in the 12-bar blues. The dominant seventh chord occurs frequently in the more traditional sequence, which starts as follows in the key of C:

| C7 | F7 | C7 | Gm7 C7 | F7 ...

The scale associated with the initial C7 is likely to include the following notes: C, D, E♭, E, F, G♭, G, A, B♭. The E♭ here is a "blue" note. The F7 chord in the second measure, however, is much less likely to contain the corresponding blue note of A♭, which is foreign to the C-major key of the chord sequence as a whole. The F7 is more likely to have the associated scale: F, G♭, G, A, B♭, B, C, D, E♭. Musicians extemporize their melodies with a feel for these harmonic nuances. They have to practice improvising to a particular chord sequence to build up a procedural knowledge of its scalic possibilities. Hence, their tacit knowledge reflects the distinction between chordal and passing notes and a sensitivity to a shift in the passing notes according to the harmonic context.

The second constraint is that musicians improvise melodies that embody a pleasing *contour*. The role of contour in the perception of tonal music has been established experimentally (see, e.g., Dowling, 1978). Roughly speaking, a musician plays a runs of notes that are fairly close to one another in

pitch, and then, for variety, introduces some larger leaps in pitch, and so on. Once again, the principles for generating a variety of contours do not call for a working memory for intermediate results. The contour determines the approximate choice of pitch, and the harmonic constraints of the current chord narrow the choice down still further. The constraints and the range of the instrument sometimes reduce the choice to a single pitch, but more often there is a small set of possibilities, and the final choice from them is made arbitrarily.

To illustrate this theory, let us consider for a final time the fragment from Bud Powell's improvisation in Figure 1. The contours for the first phrase are as follows, where "S" denotes a small step in the contour, that is, an interval of a major or minor second:

S S S | S S S S S S S S S S |

As one would expect, these steps include both notes from the chords and passing notes. The second phrase has the following contour, where "I" denotes a step of an interval of greater than a second:

| I I I I I I I I | S S S S S I S I I I |

Again as one would expect, most of the notes on I steps in the contour occur in the accompanying chord. In other words, there is a strong correlation between intervallic steps and notes from within chords, whereas passing notes tend to occur on small steps in the contour.

The principle of algorithmic demands implies that melodies can be created without intermediate results in working memory. One way in which Rich Feit and I have tested this claim is with computational modeling. We have written programs to generate jazz bass lines. Many programs exist that improvise jazz. The point of our program was to show that the task was feasible without the use of a working memory for intermediate results. Our program takes a chord sequence represented in symbols as its input—either an actual chord sequence or one generated by a program that implements the principles described earlier—and it produces as its output an appropriate bass line. It also generates a rudimentary harmonic accompaniment. The output of the program is a numerical specification of the pitch of each note, its onset and offset times, and its intensity. These numerals were used to synthesize actual base notes.

The program selects whether the next interval is large or small from a set of rules for contours derived from a corpus of bass lines. The rules are based on principles akin to those of Parsons' (1975) *Directory of Tunes and Musical Themes*, which represents any tune merely by its contour. It represents, for example, the opening theme of Beethoven's Fifth symphony as follows:

* R R D U R R D . . .

where * denotes the first note, R a repeat of the previous note, U an upward step, and D a downward step. These eight symbols are common to

five other themes, including one from Sullivan's *HMS Pinafore*. But, once the first 15 notes of any theme in the classical repertory have been encoded in the notation, it is almost always identified uniquely. The contours for improvisation assume that there are no asymmetries between rising intervals and falling intervals. This assumption was corroborated by the Parsons Directory: when one inverts the contour of the first few notes of a given entry, the result always corresponds to some other theme in the directory. The inversion of Beethoven's opening provides the contour of several themes, including one from Sullivan's *Pirates of Penzance*. Hence, the contours used in the program represent the first note of a phrase, the repeat of a previous note, a small step of a second, a chromatic run of notes, a leap of an octave, and any other intervallic step of greater than a second. The program maintains a buffer that contains only the previous note in its improvisation in order to realize any of these possibilities. The remaining constraints require only access to the current chord in the sequence, its associated scale, and the current beat of the measure.

Once the program has selected the next item in the contour, it chooses its precise pitch. If the contour calls for a repeat of the previous note, its pitch is already specified, otherwise the contour merely constrains the choice to a set of notes. The program is equipped with a specification of each of the major sorts of chord and knowledge of the acceptable passing notes that may occur with them. This knowledge associates a scale to each chord in the harmonic sequence. The result of all of these constraints is a set of notes, and so finally the program chooses randomly from among them unless, as occasionally happens, the constraints narrow the choice down to a single note.

Figure 4 presents a typical output from the program when its input was the chord sequence for *Off Minor*. The program performs at the level of a moderately competent beginner. It uses passing notes, including chromatic runs, but it makes no use of motifs other than those that occur by chance. Readers might wonder what would improve its performance. The answer is the inclusion of a richer knowledge of contours and passing notes. Yet, the program does illustrate the feasibility of the central theoretical claim of

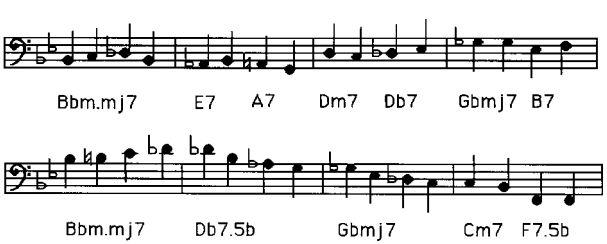


Fig. 4. A typical bass line generated by the computer program from the chord sequence of *Off Minor*.

algorithmic demands: in order to improvise, musicians need not use a working memory for the results of intermediate computations. The program makes no use of them, and its memory buffer holds only the previous note that it improvised, and its current place in the chord sequence, the contour, and the measure.

Conclusions

The NONCE definition of creativity yields three sorts of creative algorithm. The first sort uses a neo-Darwinian process of arbitrary generation followed by critical evaluation. Despite claims to the contrary, it is an open question whether anyone ever creates in this way. But, such an algorithm is far too inefficient for jazz improvisation. It yields many outputs that are not viable, and it is inconceivable for creativity in real time. The second sort of algorithm is neo-Lamarckian, and it uses the criteria of a genre to generate possibilities and then makes an arbitrary choice whenever there is more than one possibility. This sort of algorithm is feasible for the rapid creation of viable results, and it is likely to underlie jazz improvisation. The crux is that musicians can generate novel melodies without having to store intermediate results in working memory. The hypothesis is tentative. But, it is borne out by a computer program of this sort that produces passable results, and by a recent experiment (carried out by Ivor Holloway and me) in which loading up musicians' working memory had no adverse effects on improvisation. The third sort of algorithm makes multiple iterations of a stage in which criteria are used to generate ideas and a stage in which other criteria are used to evaluate them. This sort of algorithm appears to underlie many sorts of artistic and scientific creation. In the case of jazz, the creation of chord sequences for improvisation is likely to depend on such an algorithm. It calls for the retention of intermediate results in working memory, although the use of musical notation obviates the need to hold such results in actual working memory.

Jazz musicians know by heart the chord sequences on which they improvise. These sequences are consciously accessible and readily communicated. Like many tonal sequences, their composition calls for considerable computational power. Musicians also have in their heads a set of unconscious principles that control melodic improvisation. This procedural knowledge is acquired at the cost of considerable work. It embodies principles governing harmony, meter, rhythm, and contour (see also Berliner, 1994). It enables musicians to improvise in real time. The computationally most demanding process, the creation of rich chord sequences, is necessarily compositional, whereas improvisation is less demanding in computational terms. Both can be captured in rules. Hence, instead of a list of fragments

of rhythms, motifs, and so on, the algorithms described in this article make use of rules. A popular objection to this idea is that great artists often "break the rules." But if artists break the rules, then they must either make an arbitrary choice regardless of the rules, as when a musician chooses to play an entirely "random" sequence of notes, or else they must be following some other criteria. In the first case, nothing is easier than to formulate rules that allow random choices. In the second case, the new criteria can also be characterized by rules granted that the criteria can be discovered. In short, the objection appears to be groundless. The onus is on those who raise it to characterize what they mean by "breaking the rules" and why the process cannot itself be modeled in further rules.

The major innovators in jazz—from Louis Armstrong to Charlie Parker—were profoundly creative. They invented new genres of improvisation. Readers may well wonder how such innovations came about. They appear to depend on a multistage procedure that leads to a new genre. Universal constraints on the process are unlikely to exist, and so it is hard to resist concluding that the problem is computationally intractable. Any algorithm needs to be guided by some sort of constraints—not too many, and not too few—and these constraints arise from knowledge of the particular domain, but their specific nature only becomes apparent to us after the revolution.[1]

References

Bateson, G. (1979). *On mind and nature*. London: Wildwood House.

Berliner, P. F. (1994). *Thinking in jazz: The infinite art of improvisation*. Chicago: The University of Chicago Press.

Boolos, G., & Jeffrey, R. (1989). *Computability and logic* (3rd ed.). Cambridge: Cambridge University Press.

Campbell, D. T. (1960). Blind variation and selective retention in creative thought as in other knowledge processes. *Psychological Review, 67*, 380–400.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Cork, C. (1988). *Harmony by LEGO bricks: A new approach to the use of harmony in jazz improvization*. Leicester: Tadley.

Dawkins, R. (1976). *The selfish gene*. Oxford: Oxford University Press.

Dowling, W. J. (1978). Scale and contour: two components of a theory of memory for melodies. *Psychological Review, 85*, 341–354.

Forte, A. (1979). *Tonal harmony in concept and practice* (3rd ed.). New York: Holt, Rinehart, and Winston.

Freud, S. (1959). Creative writers and day-dreaming. In J. Strachey (Ed. and trans.) *Standard edition of the complete psychological works of Sigmund Freud (Vol. 9)*. London: Hogarth Press. (Original work published 1908)

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.

1. I am grateful to various colleagues for various sorts of help and criticism: Malcolm Bauer, Richard Feit, Carol Krumhansl, Jung Min Lee, and Tom Ormerod. I also thank reviewers of the article for their constructive criticisms: Geoff Collier, James L. Collier, and Jeff Pressing.

Hopcroft, J. E., & Ullman, J. D. (1979). *Formal languages and their relation to automata.* Reading, MA: Addison-Wesley.
Johnson-Laird, P. N. (1991a). Jazz improvization. In P. Howell, R. West, & I. Cross (Eds.), *Representing musical structure* (pp. 291–325). London: Academic Press.
Johnson-Laird, P. N. (1991b). Rhythm and meter: A theory at the computational level. *Psychomusicology, 10,* 88–106.
Johnson-Laird, P. N. (1993). *Human and machine thinking*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Kernfeld, B. (1988). Beat. In B. Kernfeld (Ed.) *The New Grove Dictionary of Jazz, Vol. 1: A-K* (pp. 85–88). London: Macmillan.
Koestler, A. (1964). *The act of creation.* London: Hutchinson.
Krumhansl, C. (1990). *Cognitive foundations of musical pitch*. Oxford: Oxford University Press.
Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, MA: MIT Press.
Levitt, D. A. (1981). *A melody description system for jazz improvisation.* M.Sc. thesis, Massachusetts Institute of Technology, Cambridge, MA.
Longuet-Higgins, H. C. (1962). Letter to a musical friend. *The Music Review, 23,* 244–248.
Longuet-Higgins, H. C. (1987). *Mental processes: Studies in cognitive science.* Cambridge, MA: Bradford Books, MIT Press.
Longuet-Higgins, H. C., & Lee, C. S. (1984). The rhythmic interpretation of monophonic music. *Music Perception, 1,* 424–441.
Mayr, E. (1982) *The growth of biological thought: Diversity, evolution, and inheritance.* Cambridge, MA: Belknap, Harvard University Press.
Mednick, S. A. (1962). The associative basis of the creative process. *Psychological Review*, 69, 220–232.
Miller, G. A. (1956). The magical number seven, plus or minus two. *Psychological Review*, 63, 81–97.
Parsons, D. (1975). *The dictionary of tunes and musical themes*. Cambridge: Spencer Brown.
Penrose, R. (1989). *The emperor's new mind: Concerning computers, minds, and the laws of physics*. Oxford: Oxford University Press.
Perkins, D. N. (1981). *The mind's best work*. Cambridge, MA: Harvard University Press.
Povel, D. J. (1984). A theoretical framework for rhythm perception. *Psychological Research*, *45,* 315–337.
Poincaré, H. (1929). *The foundations of science: Science and hypothesis, the value of science, science and method*. New York: The Science Press.
Power, M. J. (1986). A technique for measuring processing load during speech production. *Journal of Psycholinguistic Research, 15,* 371–382.
Pressing, J. (1987). The micro- and macrostructure of improvised music. *Music Perception, 5*, 133–172.
Rameau, J. P. (1971). *Treatise on harmony*. New York: Dover Books. (Original work published 1722)
Reber, A. S. (1985). *The Penguin dictionary of psychology*. Harmondsworth, Middlesex: Penguin.
Schuller, G. (1968). *Early jazz: Its roots and musical development*. Oxford: Oxford University Press.
Schuller, G. (1989). *The swing era: The development of jazz, 1930–1945*. Oxford: Oxford University Press.
Simonton, D. K. (1984). *Genius, creativity, and leadership: Historiometric inquiries*. Cambridge, MA: Harvard University Press.
Skinner, B. F. (1953). *Science and human behavior*. New York: Macmillan.
Steedman, M. J. (1982). A generative grammar for jazz chord sequences. *Music Perception*, *2,* 52–77.
Steedman, M. J. (1994). The blues and the abstract truth: Music and mental models. In J. Oakhill & A. Garnham (Eds.), *Mental models in cogntive science* (pp. 305–318). Hove, Sussex: Psychology Press.

Ulrich, J. W. (1977). The analysis and synthesis of jazz by computer. *Fifth International Joint Conference on Artificial Intelligence* (pp. 865–872). Hillsdale, NJ: Erlbaum.
Wallas, G. (1926). *The art of thought*. London: Jonathan Cape.
Witmer, R. (1988). Notation. In B. Kernfeld (Ed.), *The New Grove Dictionary of Jazz, Vol. 2: L-Z* (pp. 253–260). London: Macmillan.